NASA CONTRACTOR
REPORT

NASA CR-161213

RESEARCH STUDY: DEVICE TECHNOLOGY
Star Router User's Guide

By M&S Computing, Inc.
Huntsville, Alabama

February 27, 1979

Prepared for:

NASA - George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
NASA CR-161213		

4. TITLE AND SUBTITLE	5. REPORT DATE
Research Study: Device Technology Star Router User's Guide	February 1979
	6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) R. A. Wright	8. PERFORMING ORGANIZATION REPORT #
9. PERFORMING ORGANIZATION NAME AND ADDRESS M&S Computing, Inc. Huntsville, Alabama	10. WORK UNIT NO.
	11. CONTRACT OR GRANT NO. NAS8-32632
12. SPONSORING AGENCY NAME AND ADDRESS National Aeronautics and Space Administration Washington, D. C. 20546	13. TYPE OF REPORT & PERIOD COVERED Contractor Report
	14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

This work was done under the technical supervision of Mr. Jerry B. Franks, George C. Marshall Space Flight Center, Alabama.

16. ABSTRACT

The STAR Router program has been developed to perform automated layout of LSI discretionary interconnection masks. The input and output for the router are standard PR2D data files. A state-of-the-art cellular path-finding procedure, based on Lee's Algorithm, has been developed to produce fast, shortest distance routing of microcircuit net data.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
	For Early Domestic Dissemination F. BROOKS MOORE Director, Electronics and Control Lab, MSFC

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	47	NTIS

MSFC - Form 3292 (Rev. December 1972)

For sale by National Technical Information Service, Springfield, Virginia 22151

PREFACE

This report contains the user information and software specifications for the STAR Router Program. This work was performed by M&S Computing, Inc., under Contract No. NAS8-32632 for the Electronic Development Division of the George C. Marshall Space Flight Center (MSFC).

Prepared by:

R. A. Wright

Ronald E. Pitts

TABLE OF CONTENTS

1. INTRODUCTION

The STAR Router program has been developed to perform automated layout of LSI discretionary interconnection masks. The input and output for the router are standard PR2D data files. A state-of-the-art cellular path-finding procedure, based on Lee's Algorithm*, has been developed to produce fast, shortest distance routing of microcircuit net data.

*C.Y. Lee. "An Algorithm for Path Connections and Its Applications," IRE Transactions on Electronic Computers, Vol. EC-10, September, 1961, pp. 346-365.

2. DESIGN PHILOSOPHY

Current routing technology allows the designer three areas of emphasis: routing speed, distance efficiency, and physical efficiency. Lee's Algorithm was chosen as a basis for the STAR router logic because it is ideally suited to orthogonal path finding on a cellular plane. The algorithm has the intrinsic characteristic of seeking shortest distance routes through and around various types of obstacles. The STAR router enhances this algorithm with level switching logic, and has designed-in speed efficiency as well as logic to avoid routes which are physically inferior with respect to the microcircuit technology in use.

Speed efficiency was achieved in two ways: minimization of input and output to secondary storage, and routing performed completely in memory.

Input and output were kept to a minimum throughout the program. The STAR router makes only one interpretive pass through the PR2D input file. During this pass all net data is stored in memory. When the entire input file has been scanned, the accumulated data is sorted to produce a single package of information for each net; this information is then placed in an auxiliary input file. Each record of this file is read once during the routing phase. Produced route data is written to a scratch file temporarily. When routing is complete, this scratch data is merged with the input file to form the PR2D output file.

In order for the route selection logic to require only memory access, it was necessary to encode all information about a discrete position on the circuit planes in only four bits of memory. Each bit or combination of bits describes whether a signal path lies on each level of the circuit for a given position. This coding scheme results in a two-dimensional appearance for the grid, as the codes indicate a directional entry characteristic in a single plane, rather than positional occupancy on a level-by-level basis. Different codes are used for a net while it is being routed so that its segments may be distinguished from other circuit elements.

The heart of the program is the route selection phase. This phase consists of five sequences which are performed for each route. Before routing is begun for a net, the data package for the net is fetched from the auxiliary file.

Then a sequence is begun to find discontinuities in the net. One arbitrarily selected segment is coded as "home," as are all segments connected to it. The remaining portion of the net is then coded as a "target." Lee's algorithm is applied next to seek a route from home to target. First a sequence called the "fanout" is begun. This involves searching outward in a wave from home, leaving ordered codes behind as the wave front is propagated in all directions. This sequence ends when a target segment is reached. This sequence automatically eliminates most lengthy or otherwise inefficient routes, due to the ordering of the codes left behind the wave. Thus, only a few path options remain to be examined. The next sequence, called the "retrace" or "traceback," analyzes these paths by following the fanout codes in reverse order (from target to home), recording obstruction and distance information for each possible route. Logic is then exercised to find the single path with fewest corners and level changes, as well as shortest distance.

After retrace, a fourth sequence recodes the grid to its original state so that future fanout and retraces can be performed. The final sequence takes the new route and codes it into the circuit grid, then saves the route information in a scratch file (in PR2D format). During this sequence, logic is exercised to eliminate any physical inefficiencies (such as adjacent feed-throughs) in the new route.

These five sequences are repeated until the current net is completely continuous, or until no path can be found to resolve remaining discontinuities. The scratch data is then merged with the original input to form the program's output file.

3. USER INFORMATION

3.1 General Operation

To run the router, the user must build a Xerox CP-V format batch deck on cards or on disk. This deck must contain the job control language to assign the input data control block (DCB) and the control input DCB to data files. These DCB's are F:3 and F:1, respectively. The deck must also contain limit information and a run card to start the router. The following is an example of a deck which will initiate routing of the data in file XXX, using control input from ZZZ:

Card #	Image
1	!JOB
2	!LIMIT Δ (TIME,30),(CORE,72)
3	!SET Δ F:3 Δ /XXX;IN
4	!SET Δ F:1 Δ /ZZZ;IN
5	!RUN Δ (LMN,L:ROUTE)

Note that in the above example, a core limit of 72K words is specified. The program is actually about 63K words in size, and has no overlays. Also, the load module name used is L:ROUTE, but the name can be changed as desired.

The following command (from TEL) will submit the batch job for execution:

```
!BATCH NNNN
```

where NNNN is the name of the file above.

If files other than XXX and ZZZ are to be used for the job, the following command should be used:

```
!BATCH NNNN 'XXX' = 'AAA', 'ZZZ' = 'BBB'
```

where AAA is the file to use in place of XXX, and BBB is the file to use in place of ZZZ.

The file assigned to the input DCB must contain data in standard PR2D format. See Appendix E for restrictions on this data.

The control input file requires three card images as follows:

Card #	Image	Description
1	xxx,zzz	These are the x and y grid dimensions. The product may not exceed 160,000, due to memory limitations.
2	aaa,bbb	These are biases to be subtracted from x and y coordinate data.
3	nnn	This is the scale factor applied after removal of the biases.

An optional fourth card, described in Section 3.2.1, may also be included.

During execution, progress messages and error messages will indicate the status of the routing. These are described in Appendix D.

The program output will be written to I/O unit 10, unless multipass routing is invoked. In multipass routing, the output may reside on I/O unit 10 or 11. In any case, the output unit will be listed for the user on the message printout. The files for units 10 and 11 are D:10 and D:11, respectively.

3.2 Operating Techniques

3.2.1 Multipass Routing

If the user has a data set for which it is desirable to perform short distance routing before longer routing, the multipass capability of the STAR router should be used. This feature allows the user to specify a set of sequential distance limits which will be applied in consecutive complete routing passes through the net data. During each pass, only those routes which can be completed within the distance supplied will be inserted.

To activate multipass, the user must include a fourth card in the control input file. This card should contain the sequence of up to twenty limits (in whole grid units), separated by commas.

NOTE: Though this feature may produce better routes for short connections, the overall quality of the routes may be lower than that which a single pass would produce.

3.2.2 Input Editing

If a user's data set contains one or more nets which are difficult to route using single or multiple passes, the user may wish to give these nets a higher priority than the others. An effective technique for accomplishing this is editing of the PR2D input file. This involves finding the first occurrence of level six data for the difficult nets, then moving these first entries up to the top of the level. This will cause the moved nets to be routed first, often enabling complete routing of the data set which previously did not finish.

APPENDIX A

GRID CODES

The following is a list of the meaning of each of the possible numerical values of a four-bit grid position.

Hex Value	Meaning
0	Vacant position
1	Fanout code 1
2	Fanout code 2
3	Home segment code
4	Level 8 occupancy
5	Level 8 and fanout code 1
6	Level 8 and fanout code 2
7	Home code (special)
8	Level 6 occupancy
9	Level 6 and fanout code 1
A	Level 6 and fanout code 2
B	Target segment code
C	Level 6 and Level 8 occupancy
D	Level 7 occupancy (feed-thru)
E	Unused
F	Target segment

APPENDIX B

FANOUT RULES

During fanout, or the process of searching from home to target segments, several rules apply to the propagation of the wave front. They are as follows:

1. Each cell on the edge of the current wave front is eligible to propagate.

2. All cells propagated to during a scan of the wave front will form the new wave front.

3. The four cells orthogonally adjacent to the propagating cell are eligible for propagation.

4. Each level of the grid has a direction associated with it as follows: Level 6 = X-axis, Level 7 = Z-axis, Level 8 = Y-axis.

5. Propagation from or to an occupied cell may not take place along the axis associated with the occupant.

6. Cells previously propagated to are ineligible.

7. Vacant cells are always eligible.

8. Dually occupied and feed-through cells are ineligible.

9. Fanout codes are logically ordered with existing codes.

10. Fanout codes proceed in the order 21122112211...

11. If a target is encountered during a wave front scan, fanout will cease at the end of that scan.

APPENDIX C

RETRACE RULES

The retrace sequence, which proceeds from targets back to home, is guided by the codes left during fanout. The following rules apply:

1. Retrace will follow the code sequence ...22112211..., starting with the reverse of the last two codes used by fanout.

2. Propagation will proceed one cell at a time in a straight line until an incorrect code or barrier is encountered.

3. When a forced turn is produced as in rule 2, both directions of escape will be tested.

4. If more than one target cell was hit during the last fanout wave scan, separate retraces will be attempted from each target cell hit.

5. Retrace ends when a home segment is encountered.

APPENDIX D

RUN TIME MESSAGES

The following messages may appear on the run printout. Their meaning is described at right.

Error Messages	Meaning
ERROR OR END-OF-FILE WHILE READING CONTROL INPUT	Bad or missing cards were found in the control input file.
MAXIMUM GRID SIZE EXCEEDED	The product of the X and Y grid dimensions exceeds 160,000.
PREMATURE END-OF-FILE HIT ON PR2D INPUT	Input file scan did not find the end of Level 8 in the file.
GRID DIMENSIONS EXCEEDED IN RIDP	Coordinates were found in the input file which exceed the specified grid dimensions.
CORE LIMITS EXCEEDED WHILE PROCESSING PR2D INPUT	The maximum space allocated for processing input data has been overflowed. Input file is too large.
NON-ORTHOGONAL SEGMENT FOUND IN NODE XXXX.	Nonorthogonal data has been found, but the segment will be processed as a Y-axis segment.
STACK OVERFLOW IN XXXXX	A fatal overflow of available work space has occurred.
MAXIMUM NODE SIZE EXCEEDED IN XXXXX	Fatal overflow of node data storage area.
FANOUT FAILURE ON NODE XXXX	Fanout from a home segment is completely blocked. Routing continues with the next net.

Error Messages	Meaning
ROUTE FAILURE ON NODE XXXX	This message accompanies the previous one. If it occurs singly, an unexplained failure to complete a route has occurred.

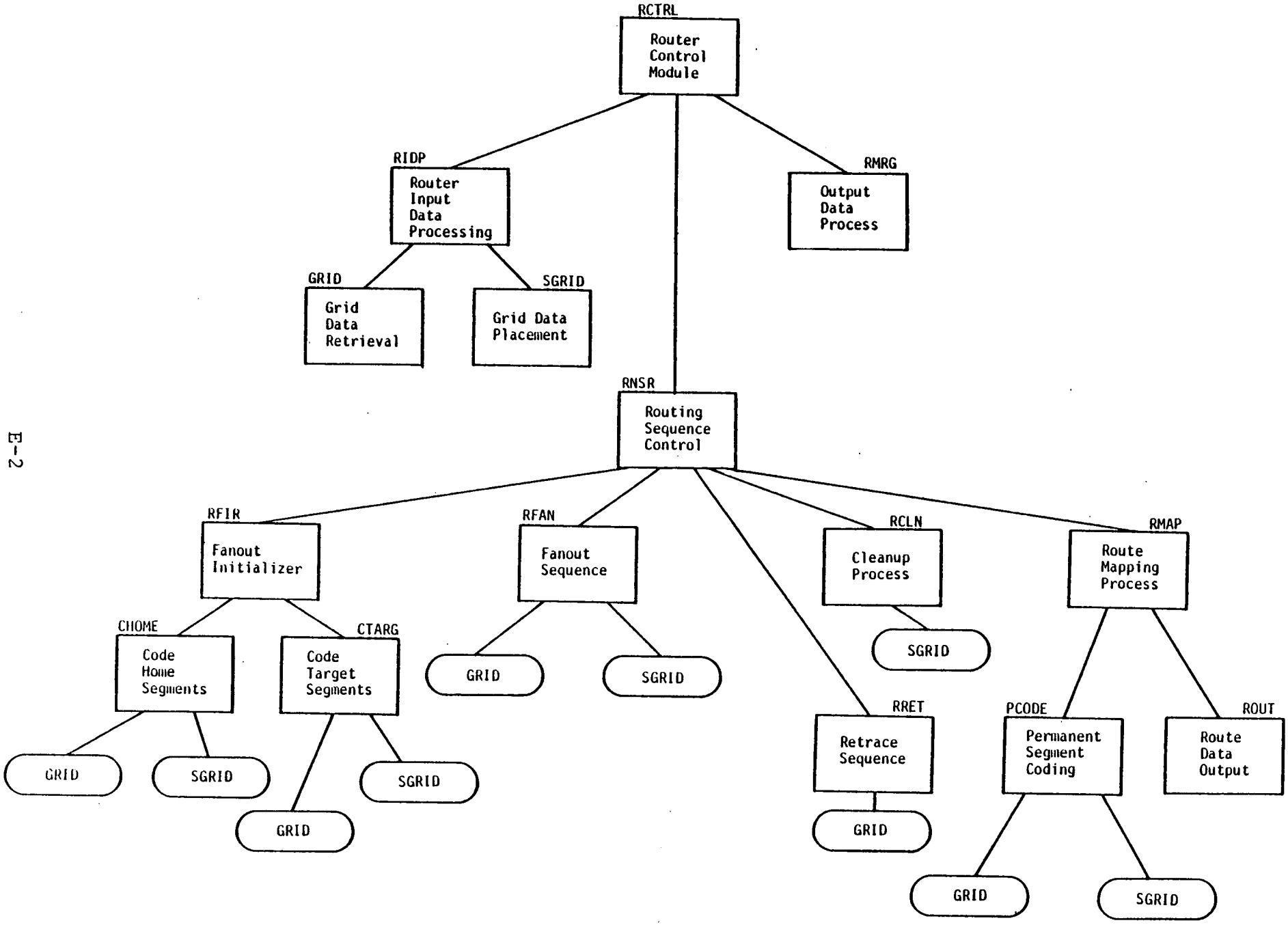
The following progress messages will normally be printed out in sequence.

```
..PROCESSING INPUT DATA
..BUILDING AUXILIARY INPUT FILE
..ROUTING PROCESS INITIATED
 FINAL AUXILIARY FILE IS ON I/O UNIT XX.
 PR2D OUTPUT FILE IS ON I/O UNIT XX.
..ROUTING COMPLETE
..BUILDING PR2D OUTPUT FILE
```

APPENDIX E

STAR ROUTER STRUCTURE

STAR ROUTER STRUCTURE



APPENDIX F

MODULE DESCRIPTIONS

08:47 FEB 13,'79 DC/PRE:RCTRL.MS03

```
**********************************************************************
*                                                                    *
*      ********      *******     *********    ********     *        *
*      *      *      *     *     *       *    *       *    *        *
*      *      *      *     *     *       *    *       *    *        *
*      ********      *     *     *       *    ********     *        *
*      *    *        *     *     *       *    *    *       *        *
*      *     *       *     *     *       *    *     *      *        *
*      *      *      *******     *********    *      *     **********
*                                                                    *
**********************************************************************
*                                                                    *
*      FUNCTION:                                                     *
*                                                                    *
*        THIS MODULE IS THE CONTROL ROUTINE FOR A 2-PLANE            *
*      PRINTED CIRCUIT ROUTER PROGRAM. IT ACQUIRES OPERATOR          *
*      ENTERED PARAMETERS AND THEN INITIATES THE ROUTING             *
*      PROCESS.                                                      *
*                                                                    *
**********************************************************************
*                                                                    *
*                                                                    *
*                                                                    *
**********************************************************************
*                                                                    *
*      C A L L I N G   S E Q U E N C E                               *
*                                                                    *
**********************************************************************
*                                                                    *
*                                                                    *
*      (STARTED BY OPERATOR AS FOLLOWS)                              *
*                                                                    *
*      !ROUTER.                                                      *
*                                                                    *
*                                                                    *
**********************************************************************
*                                                                    *
*      E X T E R N A L   R E F E R E N C E S                         *
*                                                                    *
**********************************************************************
*                                                                    *
*      MODULES THAT CALL RCTRL                                       *
*                                                                    *
*        NONE                                                        *
*                                                                    *
*      MODULES CALLED BY RCTRL                                       *
*                                                                    *
*        RIDP - PROCESSES INPUT FILE                                 *
*        RNSR - PERFORMS ROUTING SEQUENCE                            *
*        RMRG - MERGES SCRATCH OUTPUTS                               *
```

INPUT	STEP	PROCESSING	STEP	OUTPUT
	1	Call a module (RIDP) to initialize grid and process input data.	2	
	2	Set up a loop for limits on fanout. (Make a route pass for each limit.)	3	
	3	Call RNSR to perform routing function.	4	
	4	Call a module (RMRG) to build output file.	R	

08:48 FEB 13,'79 DC/PRE:RIDP.MS03

```
*****************************************************************************
*                                                                           *
*      ********     *********    ********     ********                     *
*      *       *    *       *    *       *    *       *                    *
*      *       *    *       *    *       *    *       *                    *
*      ********     *       *    *       *    ********                     *
*      *     *      *       *    *       *    *                            *
*      *      *     *       *    *       *    *                            *
*      *       *    *********    ********     *                            *
*                                                                           *
*****************************************************************************
*                                                                           *
*      FUNCTION:                                                            *
*                                                                           *
*           THIS MODULE READS THE PR2D INPUT FILE AND BUILDS AN             *
*      AUXILIARY INPUT FILE. IT ALSO PRODUCES THE ORIGINAL GRID             *
*      IMAGE USED BY THE PROGRAM.                                           *
*                                                                           *
*****************************************************************************
*                                                                           *
*                                                                           *
*****************************************************************************
*                                                                           *
*      C A L L I N G   S E Q U E N C E                                      *
*                                                                           *
*****************************************************************************
*                                                                           *
*           CALL RIDP                                                       *
*                                                                           *
*****************************************************************************
*                                                                           *
*      E X T E R N A L   R E F E R E N C E S                                *
*                                                                           *
*****************************************************************************
*                                                                           *
*      MODULES THAT CALL RIDP                                               *
*                                                                           *
*           RCTRL - ROUTER MASTER CONTROL ROUTINE                           *
*                                                                           *
*      MODULES CALLED BY RIDP                                               *
*                                                                           *
*           GRID - GRID VALUE RETRIEVAL                                     *
*           SGRID - GRID VALUE PLACEMENT                                    *
*                                                                           *
*****************************************************************************
```

INPUT	STEP	PROCESSING	STEP	OUTPUT
	/1	Rewind input and output files	2	
	2	Initiate loop for three levels: 6-8	3	I, CL (loop counter)
CL	3	Search for beginning of current level. If found, continue, else report error and stop.	4	
	4	Check for line set declaration and parameter lines and skip.	5	
	5	Begin scanning data records: all record pairs are to be coded in to the grid as segments. All node data is also saved in the node arrays.	6	AGRID, NDNM, X1, Y1, X2, Y2
	6	Repeat Step 5 until end of line set and level are encountered, then continue.	7	
I	7	If all three levels have been processed, continue. Else loop back to Step 3.	8/3	
	8	Beginning scanning node array as follows: scan down the array until a nonzero name field is found. Scan the remainder of the array, zeroing all nodes with the same name while building an output record of count-plus-data format.	9	
	9	Write the formed record to the scratch file and check for any unused names in the node array. If any are left, go back to 8. Else return.	R	

08:48 FEB 13,'79 DC/PRE:RNSR.MS03

```
*****************************************************************************
*                                                                           *
*      ********      *      *      *******     ********                    *
*      *       *     **     *     *       *    *       *                   *
*      *       *     * *    *     *            *       *                   *
*      ********      *  *   *      *******     ********                    *
*      *     *       *   * *            *     *     *                     *
*      *      *      *    **     *       *     *      *                    *
*      *       *     *     *      *******      *       *                   *
*                                                                           *
*****************************************************************************
*                                                                           *
*      FUNCTION:                                                            *
*                                                                           *
*         THIS MODULE OBTAINS RECORDS FROM AN AUXILIARY                     *
*      FILE WHICH CONTAIN NODE INFORMATION. IT THEN PERFORMS                *
*      THE SEQUENCE OF CALLS NECESSARY TO COMPLETE ALL POSSIBLE             *
*      PATH CONNECTIONS FOR EACH NODE.                                      *
*                                                                           *
*****************************************************************************
*                                                                           *
*                                                                           *
*                                                                           *
*****************************************************************************
*                                                                           *
*      C A L L I N G   S E Q U E N C E                                      *
*                                                                           *
*****************************************************************************
*                                                                           *
*                                                                           *
*        CALL RNSR                                                          *
*                                                                           *
*                                                                           *
*****************************************************************************
*                                                                           *
*      E X T E R N A L   R E F E R E N C E S                                *
*                                                                           *
*****************************************************************************
*                                                                           *
*      MODULES THAT CALL RNSR                                               *
*                                                                           *
*        RPCB - ROUTER CONTROL ROUTINE                                      *
*                                                                           *
*      MODULES CALLED BY RNSR                                               *
*                                                                           *
*        RFIR - FANOUT INITIALIZATION                                       *
*        RFAN - FANOUT SEQUENCE                                             *
*        RRET - RETRACE SEQUENCE                                            *
*        RCLN - CLEANUP SEQUENCE                                            *
*       RCODE - RECODING OF CURRENT NODE                                    *
```

INPUT	STEP	PROCESSING	STEP	OUTPUT
	1	Rewind the auxiliary file.	2	
Aux file	2	Read a pair of records from aux file to obtain a node segment list. IF EOF is encountered, go to Step 12; else continue.	3/12	CNT, NAME, CIN
CNT	3	If fewer than 2 segments are in this node, no routing will be needed, so go back to 2. Else continue.	4/2	
	4	Call a module (RFIR) to initialize for fanout.	5	DONE
DONE	5	If no connections need be made for this node, go back to Step 2. Else continue.	6/2	
	6	Call a module (RFAN) to perform fanout until a target is hit or limit is exceeded.	7	LIM
LIM	7	If a limit was exceeded in Step 6, go to Step 2, else continue.	8/2	
	8	Call a module (RRET) to perform retracing from the target back to home, selecting the optimum path.	9	OPMAP
	9	Call a module (RCLN) to clean up after fanout.	10	
OPMAP	10	Call a module (RMAP) to code in the path produced in Step 8, as well as outputing the data to scratch files for later use.	11	
	11	Go back to Step 2.	2	

INPUT	STEP	PROCESSING	STEP	OUTPUT
	12	Return.	R	

08:48 FEB 13,'79 DC/PRE:RMRG.MS03

```
**********************************************************************
*                                                                    *
*     ********     *         *    ********      *******             *
*     *        *   **       **    *         *   *         *         *
*     *        *   * *     * *    *         *   *                   *
*     ********     *  *   *  *    ********      *   *****           *
*     *    *       *   * *   *    *         *   *       *           *
*     *     *      *    *    *    *         *   *       *           *
*     *      *     *         *    *         *   *******             *
*                                                                    *
**********************************************************************
*                                                                    *
*     FUNCTION:                                                      *
*                                                                    *
*       THIS MODULE READS FILES CONTAINING SCRATCH OUTPUT            *
*     FROM THE ROUTING OPERATION AND MERGES THE INFORMATION          *
*     GENERATED WITH THE ORIGINAL INPUT FILE, FORMING THE            *
*     FINAL PRODUCT OF THE ROUTER PROGRAM.                           *
*                                                                    *
**********************************************************************
*                                                                    *
*                                                                    *
**********************************************************************
*                                                                    *
*       C A L L I N G   S E Q U E N C E                              *
*                                                                    *
**********************************************************************
*                                                                    *
*                                                                    *
*     CALL RMRG                                                      *
*                                                                    *
*                                                                    *
**********************************************************************
*                                                                    *
*       E X T E R N A L   R E F E R E N C E S                        *
*                                                                    *
**********************************************************************
*                                                                    *
*     MODULES THAT CALL RMRG                                         *
*                                                                    *
*       RPCB - ROUTER CONTROL ROUTINE                                *
*                                                                    *
*     MODULES CALLED BY RMRG                                         *
*                                                                    *
*       NONE                                                         *
*                                                                    *
**********************************************************************
```

INPUT	STEP	PROCESSING	STEP	OUTPUT
	1	Rewind scratch files and input file.	2	
	2	Loop to perform the merging operation for three levels of data.	3	
	3	Read a record from the original input file and write it to the output file.	4	
	4	If the appropriate "end level" card is found, go to Step 5. Else go to Step 3.		
	5	Continue reading cards from the input file until an "end line set" card is found; hold this card until later, and proceed to Step 6.	6	
	6	Transfer all cards from the scratch file for the current level to the output file.	7	
	7	Write the saves cards (S5) to the output file. If any levels remain to be merged, go back to Step 3. Else copy the remaining cards from the input file to the output file and return.	3/R	

08:47 FEB 13,'79 DC/PRE:RFIR.MS03

```
**********************************************************************
*                                                                    *
*      ********     *********    *********    ********              *
*      *       *    *            *           *       *             *
*      *       *    *            *           *       *             *
*      ********     *******      *           ********              *
*      *    *       *            *           *    *                *
*      *     *      *            *           *     *               *
*      *      *     *            *********   *      *              *
*                                                                    *
**********************************************************************
*                                                                    *
*      FUNCTION:                                                     *
*                                                                    *
*        THIS MODULE TAKES THE CURRENT NODE ARRAY AND SCANS IT,      *
*      CODING THE SEGMENTS AS HOMES OR TARGETS, AND PRODUCES THE     *
*      INITIAL FANOUT ARRAY.                                         *
*                                                                    *
**********************************************************************
*                                                                    *
*                                                                    *
**********************************************************************
*                                                                    *
*      C A L L I N G   S E Q U E N C E                               *
*                                                                    *
**********************************************************************
*                                                                    *
*                                                                    *
*        CALL RFIR                                                   *
*                                                                    *
**********************************************************************
*                                                                    *
*      E X T E R N A L   R E F E R E N C E S                         *
*                                                                    *
**********************************************************************
*                                                                    *
*      MODULES THAT CALL RFIR                                        *
*                                                                    *
*        RNSR - ROUTING SEQUENCE CONTROL                             *
*                                                                    *
*      MODULES CALLED BY RFIR                                        *
*                                                                    *
*        GRID - GRID VALUE RETRIEVAL                                 *
*       CHOME - CODES SEGMENTS AS HOMES                              *
*       CTARG - CODES SEGMENTS AS TARGETS                            *
*                                                                    *
**********************************************************************
```

INPUT	STEP	PROCESSING	STEP	OUTPUT
	1	If we have been here before for this fanout, go to Step 7. Else continue.	2/7	
	2	Initialize stack and segment test flags.	3	
	3	Use the stacked segment as home. If non left, go to Step 7.	4/7	
	4	Start a loop to scan all the segments to see if they intersect the home segment.	5	
	5	Get the next segment, and test it. If it intersects home, go to Step 6. Else loop back to Step 3.	6/3	
	6	Flag this segment as a part of home and place it on the stack for testing. Go back to Step 5 until all segments are checked. Then go to 3.		
	7	Loop for each segment of the current node. If it is home, code it as such. Same for targets (anything that is not home); modules CHOME and CTARG are called for these purposes.	8	
	8	If no targets found call RCODE, then return.	R	

08:47 FEB 13, '79 DC/PRE:RFAN.MS03

```
*************************************************************************
*                                                                       *
*     ********   *********          *          *     *                  *
*     *       *  *                 * *         **    *                  *
*     *       *  *                *   *        * *   *                  *
*     ********   *******         *******       *  *  *                  *
*     *    *     *              *       *      *   * *                  *
*     *     *    *              *       *      *    **                  *
*     *      *   *              *       *      *     *                  *
*                                                                       *
*************************************************************************
*                                                                       *
*    FUNCTION:                                                          *
*                                                                       *
*        THIS MODULE PERFORMS THE FANOUT SEQUENCE FOR THE               *
*    ROUTER PROGRAM.  THE FANOUT IS A CODING SEQUENCE THAT              *
*    STARTS AT HOME SEGMENTS AND EXPANDS INA WAVE IN ALL                *
*    DIRECTIONS UNTIL AT LEAST ONE TARGET HAS BEEN REACHED.             *
*    THE HITS ARE RECORDED FOR RETRACING.                               *
*                                                                       *
*************************************************************************
*                                                                       *
*                                                                       *
*                                                                       *
*************************************************************************
*                                                                       *
*     C A L L I N G   S E Q U E N C E                                   *
*                                                                       *
*************************************************************************
*                                                                       *
*                                                                       *
*         CALL RFAN                                                     *
*                                                                       *
*                                                                       *
*************************************************************************
*                                                                       *
*     E X T E R N A L   R E F E R E N C E S                             *
*                                                                       *
*************************************************************************
*                                                                       *
*    MODULES THAT CALL RFAN                                             *
*                                                                       *
*       RNSR - ROUTING SEQUENCE CONTROL                                 *
*                                                                       *
*    MODULES CALLED BY RFAN                                             *
*                                                                       *
*       GRID - GRID VALUE RETRIEVAL                                     *
*      SGRID - GRID VALUE PLACEMENT                                     *
*                                                                       *
*************************************************************************
```

INPUT	STEP	PROCESSING	STEP	OUTPUT
FCNT	1	Start a loop for each member of the fanout wave front.	2	
GRID, FAN	2	Check each of the four adjacent cells for fanout eligibility, and code them where eligible. Add coded nodes to new fanout wavefront.	3	FAN
HIT	3	If any target hits were found in Step 2, save them. Repeat 2 for the loop established in Step 1.	2/4	HIT, HSTK
HIT	4	If any hits were saved by by Steps 2 and 3, go to Step 5. Else repeat 1 to 3 for new wavefront. If new wave-front is blocked completely, go to Step 6.	1/5/ 6	
HIT	5	Hits were found during last fanout step, so return with successful status.	R	LIM
	6	Fanout is limited or blocked, so return no success status.	R	LIM

08:48 FEB 13,'79 DC/PRE:RRET.MS03

```
**********************************************************************
*                                                                    *
*      ********     ********     *********    *********             *
*      *      *     *      *     *            *                     *
*      *      *     *      *     *            *                     *
*      ********     ********     *******      *                     *
*      *    *       *    *       *            *                     *
*      *     *      *     *      *            *                     *
*      *      *     *      *     *********    *                     *
*                                                                    *
**********************************************************************
*                                                                    *
*      FUNCTION:                                                     *
*                                                                    *
*         THIS MODULE PERFORMS THE RETRACE SEQUENCE                  *
*      FOR THE ROUTER PROGRAM. IT USES THE GRID IMAGE                *
*      BUILT BY THE FANOUT ROUTINE AND TRACES THE PATHS              *
*      FROM TARGETS TO HOME, AND SELECTS THE OPTIMUM.                *
*                                                                    *
**********************************************************************
*                                                                    *
*                                                                    *
**********************************************************************
*                                                                    *
*      C A L L I N G   S E Q U E N C E                               *
*                                                                    *
**********************************************************************
*                                                                    *
*                                                                    *
*        CALL RRET                                                   *
*                                                                    *
**********************************************************************
*                                                                    *
*      E X T E R N A L   R E F E R E N C E S                         *
*                                                                    *
**********************************************************************
*                                                                    *
*      MODULES THAT CALL RRET                                        *
*                                                                    *
*        RNSR - ROUTING SEQUENCE CONTROL                             *
*                                                                    *
*      MODULES CALLED BY RRET                                        *
*                                                                    *
*        GRID - GRID ARRAY VALUE RETRIEVAL                           *
*                                                                    *
**********************************************************************
```

INPUT	STEP	PROCESSING	STEP	OUTPUT
HSTK,	1	Initialize tree stack with hit points from fanout.	2	PSTP, PSTK
PSTK, PSTP	2	Get next tree point from the stack. If stack is empty, leave.	3	
GRID, TRMAP, TRCNT	3	Start at the current tree point and its associated code, direction, and trace map position; and proceed with retracing.	4	
GRID	4	If a barrier is encountered, stack the point as a tree point, then turn right or left and proceed.	5	PSTK, PSTP, TRCNT, TRMAP
GRID, TRMAP, OPMAP	5	If a home is encountered, check the current trace map against the optimum map and save better one.	6	TRMAP, OPMAP
	6	Continue tracing in a straight line until either 4 or 5 applies, dropping back to Step 2 when blocked completely.	3/2	

```
:47 FEB 13,'79 DC/PRE:RCLN.MS03
*************************************************************************
                                                                        *
    ********     *******    *            *     *                        *
    *       *  *       *    *            **    *                        *
    *       *  *            *            * *   *                        *
    ********   *            *            *  *  *                        *
    *    *     *            *            *   * *                        *
    *     *    *       *    *            *    **                        *
    *      *    *******     *********    *     *                        *
                                                                        *
*************************************************************************
                                                                        *
    FUNCTION:                                                           *
                                                                        *
      THIS MODULE RESTORES NORMAL CODES TO THE GRID ARRAY               *
   AFTER THE FANOUT/RETRACE SEQUENCE IS PERFORMED.                      *
                                                                        *
*************************************************************************
                                                                        *
                                                                        *
                                                                        *
*************************************************************************
                                                                        *
     C A L L I N G   S E Q U E N C E                                    *
                                                                        *
*************************************************************************
                                                                        *
                                                                        *
      CALL RCLN                                                         *
                                                                        *
                                                                        *
*************************************************************************
                                                                        *
     E X T E R N A L   R E F E R E N C E S                              *
                                                                        *
*************************************************************************
                                                                        *
    MODULES THAT CALL RCLN                                              *
                                                                        *
      RNSR - ROUTING SEQUENCE CONTROL                                   *
                                                                        *
    MODULES CALLED BY RCLN                                              *
                                                                        *
      GRID - GRID VALUE RETRIEVAL                                       *
     SGRID - GRID VALUE PLACEMENT                                       *
                                                                        *
*************************************************************************
```

INPUT	STEP	PROCESSING	STEP	OUTPUT
XMIN, XMAX, YMIN, YMAX, GRID	1	Scan through all areas of the grid altered by fanout, restoring codes to normal form.	2	GRID
ISTK, ISTP	2	Recode all cells whose values were saved on the stack by CHOME and CTARG.	R	GRID

08:48 FEB 13,'79 DC/PRE:RMAP.MS03

```
**********************************************************************
*                                                                    *
*     ********     *        *        *        ********               *
*     *      *     **      **      * *       *      *               *
*     *      *     * *    * *      *  *      *      *               *
*     ********     *  *  *  *      *   *     ********               *
*     *    *       *   **   *      *****     *                      *
*     *     *      *        *     *     *    *                      *
*     *      *     *        *    *       *   *                      *
*                                                                    *
**********************************************************************
*                                                                    *
*     FUNCTION:                                                      *
*                                                                    *
*        THIS MODULE CODES IN THE OPTIMUM RETRACE PATH SELECTED      *
*     BY THE FANOUT/RETRACE SEQUENCE, AND OUTPUTS THE GENERATED      *
*     INFORMATION TO SCRATCH FILES.                                  *
*                                                                    *
**********************************************************************
*                                                                    *
*                                                                    *
**********************************************************************
*                                                                    *
*     C A L L I N G   S E Q U E N C E                                *
*                                                                    *
**********************************************************************
*                                                                    *
*                                                                    *
*        CALL RMAP                                                   *
*                                                                    *
**********************************************************************
*                                                                    *
*     E X T E R N A L   R E F E R E N C E S                          *
*                                                                    *
**********************************************************************
*                                                                    *
*     MODULES THAT CALL RMAP                                         *
*                                                                    *
*       RNSR - ROUTING SEQUENCE CONTROL                              *
*                                                                    *
*     MODULES CALLED BY RMAP                                         *
*                                                                    *
*       GRID - GRID VALUE RETRIEVAL                                  *
*       ROUT - OUTPUT OF GENERATED ROUTE INFORMATION                 *
*      PCODE - PERMANENT CODING OF SEGMENTS                          *
*                                                                    *
**********************************************************************
```

INPUT	STEP	PROCESSING	STEP	OUTPUT
OPMAP, OPCNT	1	Check to make sure a route was found. If none, return error status.	2/R	DONE
OPMAP, GRID	2	Determine whether to place feed-throughs on route end points.	3	J,K
OPCNT, J,K	3	Start a loop to code in feed-throughs.	4	
	4	If feed-throughs are required and no adjacent, code them in. Repeat this step as determined by Step 3.	4/5	GRID
OPCNT, OPMAP	5	Loop to place all segments in the new route into the grid and net data.	R	GRID

08:47 FEB 13,'79 DC/PRE:CHOME.MS03

```
*******************************************************************************
*                                                                             *
*      *******     *       *     *******     *       *     *********        *
*     *       *    *       *    *       *    **     **     *                 *
*     *            *       *    *       *    * *   * *     *                 *
*     *            *********    *       *    *  * *  *     *******           *
*     *            *       *    *       *    *   *   *     *                 *
*     *       *    *       *    *       *    *       *     *                 *
*      *******     *       *     *******     *       *     *********        *
*                                                                             *
*******************************************************************************
*                                                                             *
*     FUNCTION:                                                               *
*                                                                             *
*       THIS MODULE CODES SEGMENTS AS HOME IN THE GRID ARRAY.                 *
*                                                                             *
*******************************************************************************
*                                                                             *
*                                                                             *
*                                                                             *
*******************************************************************************
*                                                                             *
*     C A L L I N G   S E Q U E N C E                                         *
*                                                                             *
*******************************************************************************
*                                                                             *
*                                                                             *
*       CALL CHOME(X1,Y1,X2,Y2)                                               *
*                                                                             *
*       WHERE X1,X2,Y1, AND Y2 ARE COORDINATES                                *
*        DESCRIBING THE SEGMENT                                               *
*                                                                             *
*                                                                             *
*******************************************************************************
*                                                                             *
*     E X T E R N A L   R E F E R E N C E S                                   *
*                                                                             *
*******************************************************************************
*                                                                             *
*     MODULES THAT CALL CHOME                                                 *
*                                                                             *
*       RFIR - FANOUT INITIALIZATION                                          *
*                                                                             *
*     MODULES CALLED BY CHOME                                                 *
*                                                                             *
*       GRID - GRID VALUE RETRIEVAL                                           *
*      SGRID - GRID VALUE PLACEMENT                                           *
*                                                                             *
*******************************************************************************
```

INPUT	STEP	PROCESSING	STEP	OUTPUT
X1, X2, Y1, Y2	1	Save grid values of segment	2	ISTK ISTP
X1, X2, Y1, Y2	2	Scan along the segment, coding in appropriate values to designate the segment as home.	3	GRID
X, Y	3	Place each value of the home segment into the initial fanout array.	R	FAN, FCNT

CHOME HIPO

08:47 FEB 13,'79 DC/PRE:CTARG.MS03

```
**********************************************************************
*                                                                    *
*      *******     *********        *       ********     *******    *
*     *       *   *         *      * *     *        *   *       *   *
*     *           *         *     *   *    *        *   *           *
*     *           *         *    *******   *********   *  *****    *
*     *           *         *   *       *  *        *  *       *   *
*     *       *   *         *   *       *  *        *  *       *   *
*      *******     *         *  *         * *        *  *******    *
*                                                                    *
**********************************************************************
*                                                                    *
*     FUNCTION:                                                      *
*                                                                    *
*       THIS MODULE CODES SEGMENTS AS TARGETS IN THE GRID ARRAY.     *
*                                                                    *
**********************************************************************
*                                                                    *
*                                                                    *
*                                                                    *
**********************************************************************
*                                                                    *
*     C A L L I N G   S E Q U E N C E                                *
*                                                                    *
**********************************************************************
*                                                                    *
*                                                                    *
*       CALL CTARG(X1,Y1,X2,Y2)                                      *
*                                                                    *
*       WHERE X1,X2,Y1, AND Y2 ARE COORDINATES                       *
*        DESCRIBING THE SEGMENT                                      *
*                                                                    *
*                                                                    *
**********************************************************************
*                                                                    *
*     E X T E R N A L   R E F E R E N C E S                          *
*                                                                    *
**********************************************************************
*                                                                    *
*     MODULES THAT CALL CTARG                                        *
*                                                                    *
*       RFIR - FANOUT INITIALIZATION                                 *
*                                                                    *
*     MODULES CALLED BY CTARG                                        *
*                                                                    *
*       GRID - GRID VALUE RETRIEVAL                                  *
*      SGRID - GRID VALUE PLACEMENT                                  *
*                                                                    *
**********************************************************************
```

INPUT	STEP	PROCESSING	STEP	OUTPUT
X1, X2, Y1, Y2	1	Save grid values of the segment.	2	ISTK, ISTP
X1, X2, Y1, Y2	2	Scan along the segment, placing codes to desginate target cells.	R	GRID

08:48 FEB 13,'79 DC/PRE:ROUT.MS03

```
*****************************************************************************
*                                                                           *
*      ********      *******     *       *    *********                    *
*      *       *     *           *       *    *                            *
*      *       *     *           *       *    *                            *
*      ********      *           *       *    *                            *
*      *    *        *           *       *    *                            *
*      *     *       *           *       *    *                            *
*      *      *      *******     *******      *                            *
*                                                                           *
*****************************************************************************
*                                                                           *
*     FUNCTION:                                                             *
*                                                                           *
*        THIS MODULE RECORDS PATH INFORMATION PRODUCED BY THE               *
*     ROUTING SEQUENCE IN SCRATCH DISK FILES.                               *
*                                                                           *
*****************************************************************************
*                                                                           *
*                                                                           *
*                                                                           *
*****************************************************************************
*                                                                           *
*      C A L L I N G   S E Q U E N C E                                      *
*                                                                           *
*****************************************************************************
*                                                                           *
*                                                                           *
*        CALL ROUT(X1,Y1,X2,Y2)                                             *
*                                                                           *
*        WHERE X1,X2,Y1,Y2 DESCRIBE A SEGMENT TO BE SAVED.                  *
*                                                                           *
*                                                                           *
*****************************************************************************
*                                                                           *
*      E X T E R N A L   R E F E R E N C E S                                *
*                                                                           *
*****************************************************************************
*                                                                           *
*      MODULES THAT CALL ROUT                                               *
*                                                                           *
*        RMAP - ROUTER MAPPING AND OUTPUT                                   *
*                                                                           *
*      MODULES CALLED BY ROUT                                               *
*                                                                           *
*        NONE                                                               *
*                                                                           *
*****************************************************************************
```

INPUT	STEP	PROCESSING	STEP	OUTPUT
X1, X2, Y1, Y2	1	Determine the level (and I/O unit) on which to write the data.	2	LVL
X1, X2, Y1, Y2, SCALE, BX, BY	2	Convert the in-core data to original units with scale and biases.	3	XX1, XX2, YY1, YY2
LVL, XX1, XX2, YY1, YY2	3	Write the segment data to the chosen I/O unit in the output format.	R	

08:47 FEB 13,'79 DC/PRE:PCODE.MS03

```
*****************************************************************************
*                                                                           *
*      ********      *******      *******      ********      *********      *
*      *      *      *     *      *     *      *      *      *              *
*      *      *      *     *      *     *      *      *      *              *
*      ********      *     *      *     *      *      *      *******        *
*      *             *     *      *     *      *      *      *              *
*      *             *     *      *     *      *      *      *              *
*      *             *******      *******      ********      *********      *
*                                                                           *
*****************************************************************************
*                                                                           *
*      FUNCTION:                                                            *
*                                                                           *
*        THIS MODULE PERMANENTLY CODES A SEGMENT INTO THE                   *
*    GRID ARRAY.                                                            *
*                                                                           *
*****************************************************************************
*                                                                           *
*                                                                           *
*****************************************************************************
*                                                                           *
*      C A L L I N G   S E Q U E N C E                                      *
*                                                                           *
*****************************************************************************
*                                                                           *
*                                                                           *
*        CALL PCODE(X1,Y1,X2,Y2)                                            *
*                                                                           *
*        WHERE X1,X2,Y1,Y2 DESCRIBE THE SEGMENT                             *
*                                                                           *
*****************************************************************************
*                                                                           *
*      E X T E R N A L   R E F E R E N C E S                                *
*                                                                           *
*****************************************************************************
*                                                                           *
*      MODULES THAT CALL PCODE                                              *
*                                                                           *
*        RMAP - ROUTE MAPPING AND OUTPUT                                    *
*                                                                           *
*      MODULES CALLED BY PCODE                                              *
*                                                                           *
*        GRID - GRID VALUE RETRIEVAL                                        *
*       SGRID - GRID VALUE PLACEMENT                                        *
*                                                                           *
*****************************************************************************
```

INPUT	STEP	PROCESSING	STEP	OUTPUT
X1, Y1, X2, Y2	1	Add the segment to the current node.	2	CIN, CNT
X1, X2, Y1, Y2	2	Add each cell of the new segment to the grid.	R	GRID

```
****************************************************************************
*                                                                          *
*      *******     *******     ********    *********   ********           *
*     *       *   *       *   *       *       *       *       *          *
*     *           *           *       *       *       *       *          *
*      *******    *   *****   ********        *       *       *          *
*             *   *       *   *    *          *       *       *          *
*     *       *   *       *   *     *         *       *       *          *
*      *******     *******    *       *   *********   ********           *
*                                                                          *
****************************************************************************
*                                                                          *
*     FUNCTION:                                                            *
*                                                                          *
*       THIS MODULE PLACES    FOUR BIT CHUNKS OF GRID DATA.                *
*                                                                          *
****************************************************************************
*                                                                          *
*                                                                          *
*                                                                          *
****************************************************************************
*                                                                          *
*      C A L L I N G   S E Q U E N C E                                     *
*                                                                          *
****************************************************************************
*                                                                          *
*                                                                          *
*       CALL SGRID(X,Y,V)                                                  *
*                                                                          *
*       WHERE X AND Y ARE GRID COORDINATES, AND V IS                       *
*        THE VALUE TO BE PLACED.                                           *
*                                                                          *
*                                                                          *
****************************************************************************
*                                                                          *
*      E X T E R N A L   R E F E R E N C E S                               *
*                                                                          *
****************************************************************************
*                                                                          *
*     MODULES THAT CALL SGRID                                              *
*                                                                          *
*       <CALLED BY MOST ROUTINES>                                          *
*                                                                          *
*     MODULES CALLED BY SGRID                                              *
*                                                                          *
*       NONE                                                               *
*                                                                          *
****************************************************************************
```

INPUT	STEP	PROCESSING	STEP	OUTPUT
X,Y	1	Compute from coordinates the word offset into the grid array, and bit offset into word.	2	I,J,K
J,K AGRID	2	Obtain the word with old data masked off.	3	L
Z,K,L, J	3	Set the new value into position and replace the word in the grid array.	R	AGRID

08:47 FEB 13,'79 DC/PRE:GRID.MS03

```
**********************************************************************
*                                                                    *
*      *******     ********    *********    ********                 *
*      *     *     *      *    *       *    *      *                 *
*      *           *      *    *       *    *      *                 *
*      *   *****   ********    *       *    *      *                 *
*      *       *   *    *      *       *    *      *                 *
*      *       *   *     *     *       *    *      *                 *
*      *******    *      *    *********    ********                 *
*                                                                    *
**********************************************************************
*                                                                    *
*    FUNCTION:                                                       *
*                                                                    *
*      THIS MODULE IS FUNCTION SUBROUTINE USED TO RETRIEVE           *
*    FOUR BIT CHUNKS OF GRID CODE DATA.                              *
*                                                                    *
**********************************************************************
*                                                                    *
*                                                                    *
**********************************************************************
*                                                                    *
*    C A L L I N G   S E Q U E N C E                                 *
*                                                                    *
**********************************************************************
*                                                                    *
*                                                                    *
*      GRID(X,Y)                                                     *
*                                                                    *
*      WHERE X AND Y ARE THE GRID COORDINATES                        *
*                                                                    *
**********************************************************************
*                                                                    *
*    E X T E R N A L   R E F E R E N C E S                           *
*                                                                    *
**********************************************************************
*                                                                    *
*    MODULES THAT CALL GRID                                          *
*                                                                    *
*      <CALLED BY MOST ROUTINES>                                     *
*                                                                    *
*    MODULES CALLED BY GRID                                          *
*                                                                    *
*      NONE                                                          *
*                                                                    *
**********************************************************************
```

INPUT	STEP	PROCESSING	STEP	OUTPUT
X,Y	1	Compute from coordinates the word offset into the grid array, and bit offset into word.	2	I,J,K
J,K AGRID	2	Obtain the word and mask off undesirable portion.	3	L
L,J	3	Right justify the resulting data and place into the output variable.	R	GRID

☆ U.S. GOVERNMENT PRINTING OFFICE 1979—640-081/583 REGION NO. 4